DKEC: Domain Knowledge Enhanced Multi-Label Classification for Diagnosis Prediction

Xueren Ge and **Abhishek Satpathy** and **Ronald Dean Williams**
and **John A. Stankovic** and **Homa Alemzadeh**
University of Virginia, Charlottesville, VA 22903 USA
{zar8jw, cqa3ym, rdw, jas9f, ha4d}@virginia.edu

Abstract

Multi-label text classification (MLTC) tasks in the medical domain often face the long-tail label distribution problem. Prior works have explored hierarchical label structures to find relevant information for few-shot classes, but mostly neglected to incorporate external knowledge from medical guidelines. This paper presents DKEC, **D**omain **K**nowledge **E**nhanced **C**lassification for diagnosis prediction with two innovations: (1) automated construction of heterogeneous knowledge graphs from external sources to capture semantic relations among diverse medical entities, (2) incorporating the heterogeneous knowledge graphs in few-shot classification using a label-wise attention mechanism. We construct DKEC using three online medical knowledge sources and evaluate it on a real-world Emergency Medical Services (EMS) dataset and a public electronic health record (EHR) dataset. Results show that DKEC outperforms the state-of-the-art label-wise attention networks and transformer models of different sizes, particularly for the few-shot classes. More importantly, it helps the smaller language models achieve comparable performance to large language models.

1 Introduction

Automated diagnosis prediction (Ma et al., 2017) is the challenging task of classifying different diseases based on a patient's EHR for applications such as treatment recommendation (e.g., selecting EMS protocols (Shu et al., 2019; Jin et al., 2023; Weerasinghe et al., 2024)) or medical billing (e.g., assigning ICD-9 codes) (O'malley et al., 2005).

Diagnosis prediction based on the free-text medical notes is known as multi-label text classification (MLTC) (Liu et al., 2017), which is the task of assigning the most relevant labels to a text instance. MLTC is more complex than the traditional multi-class text classification because the number of possible *label combinations grows exponentially* with the number of classes (Chen et al., 2017). Another challenge in diagnosis prediction is the imbalanced distribution of diagnoses as some medical conditions happen more frequently than others, causing a *long-tail data distribution*. For example, the total number of chest pain-related reports in a real-world EMS dataset is ten times more than overdose/poisoning reports (Kim et al., 2021). Training on such imbalanced datasets, also called "power-law datasets" (Rubin et al., 2012), introduces bias in model predictions towards *head* label classes while ignoring the *few-shot* or *tail* classes.

Most existing diagnosis prediction solutions (Rasmy et al., 2021; Lee et al., 2020) are task-agnostic and rely on integrating biomedical domain knowledge with transformer models in the pre-training stage. For example, CORe-BERT (van Aken et al., 2021b) uses clinical outcome pre-training to learn relations among symptoms, risk factors and clinical outcomes by incorporating Wikipedia and PubMed knowledge bases. Recent pre-trained large language models (LLMs) (Yang et al., 2022a; Luo et al., 2022; Bolton et al., 2024) have demonstrated superior performance by leveraging large external clinical corpora and huge number of parameters. However, these models only incorporate uncurated knowledge in pre-training, neglect task-specific knowledge and label relations, and are costly to fine-tune and deploy on resource-constrained devices (Jin et al., 2023; Weerasinghe et al., 2024).

To solve the class-imbalance problem in MLTC, the convolutional attention network and its variants (Kim, 2014; Li and Yu, 2020; Liu et al., 2021) were proposed to extract meaningful document representations that cover different ranges of clinical text. Other works (Rios and Kavuluru, 2018; Wang et al., 2022) integrated hierarchical information by graph convolutional neural networks to select label-

Models	Encoder	Attention Mechanism	Knowledge Integration	Knowledge Source	Datasets
(van Aken et al., 2021b)	BERT	Self-Attention	Pre-training	Wikipedia, PubMed	MIMIC-III
(Yang et al., 2022b)	MegatronBERT	Self-Attention	Pre-training	Wikipedia, PubMed	MIMIC-III
(Bolton et al., 2024)	GPT2	Self-Attention	Pre-training	PubMed	MedMCQA
(Mullenbach et al., 2018)	CNN	Label-wise Attention	ICD-9 hierarchy graph	ICD-9 description	MIMIC-III
(Rios and Kavuluru, 2018)	CNN	Label-wise Attention			MIMIC-III
(Li and Yu, 2020)	Multi-filter residual CNN	Label-wise Attention			MIMIC-III
(Zhou et al., 2021)	Multi-filter CNN	Shared Interactive Attention			MIMIC-III
DKEC (Ours)	Multi-filter CNN, Transformers	Label-wise Attention	Heterogeneous graph	Wikipedia, MayoClinic, ODEMSA	MIMIC-III & EMS

Table 1: Summary of previous works on diagnosis prediction.

relevant features. Some follow-up studies (Lu et al., 2020; Cao et al., 2020; Zhou et al., 2021) have also proposed to incorporate label co-occurrence graphs, along with hierarchical structures to capture label concurrent and mutual exclusive relations for ICD-9 code classification. However, most of these works neglected the potential benefits of incorporating *expert knowledge from medical guidelines*. External domain knowledge can provide additional information for training with few-shot labels to compensate for data scarcity and model size or be applied as constraints in training based on *label relations*.

This paper presents DKEC (Figure 1), a *knowledge and data-driven* approach to class-imbalanced MLTC by (i) automated extraction of *label-specific semantic relations* from *online sources* and (ii) integrating them as *heterogeneous knowledge graphs* with different encoders using a *label-wise attention mechanism*. Our contributions are as follows:

- We develop a method for automated construction of heterogeneous knowledge graphs from online sources (e.g., Wikipedia, MayoClinic, ODEMSA) that accurately captures semantic relations among diverse medical entities (e.g., symptoms and diseases, diseases and treatments), by medical entity extraction using chain-of-thought prompting with GPT-4 and UMLS medical concept normalization.

- We design a heterogeneous label-wise attention mechanism based on graph transformers that captures the diagnosis co-occurrence relations based on relevant medical entities in the knowledge graph and is combined with different encoders (e.g., Multi-filter CNN, BERT) to improve multi-label classification.

- We conduct extensive experiments to evaluate DKEC by applying it to language models of varying sizes using a real-world EMS dataset (Kim et al., 2021) and the MIMIC-III dataset (Johnson et al., 2016). Results show that DKEC outperforms state-of-the-art by 3.7% and 2.1% in overall top-K recall for the EMS and MIMIC-III datasets, respectively, and enhances small language model performance in few-shot classes by 10.5% and 6%.

2 Related Work

Pre-trained Transformers for Diagnosis Prediction One avenue explored in prior work is focused on large-scale pre-training from clinical admissions, discharge summaries, and other biomedical texts, such as BioBERT (Lee et al., 2020), COReBERT (van Aken et al., 2021a) and GatorTron (Yang et al., 2022a) (Table 1). Recently, it has been shown that pre-trained LLMs, including BioGPT (Luo et al., 2022) and BioMedLM (Bolton et al., 2024), can outperform general-purpose models and compete with expert-designed, domain-specific model architectures. Unlike these works, which focus on integrating external knowledge corpora for task-agnostic pre-training, we aim to incorporate task-specific knowledge and disease-related relations during the fine-tuning stage.

Label-wise Attention Networks Another line of research has focused on developing attention mechanisms to select the most relevant clinical segments for each label (see Table 1). CAML (Mullenbach et al., 2018) was the first that proposed to integrate the semantic meanings of the labels by assigning label-wise attention weights to medical text. In (Rios and Kavuluru, 2018; Vu et al., 2020), the hierarchical structure of labels was modeled and further concatenated into text features for classification. Recent studies have proposed different modules, including multiple graph aggregation (Lu et al., 2020), interactive shared representation network (Zhou et al., 2021), and hyperbolic and co-graph representation learning module (Cao et al., 2020) to capture label co-occurrence along with label hierarchy for ICD code classification. However, these works ignore the domain knowledge from other sources (e.g., medical guidelines), which can provide additional information for training with rare classes and compensate for data scarcity. Also, most of them only focused on ICD coding using convolutional neural networks (CNNs) in the

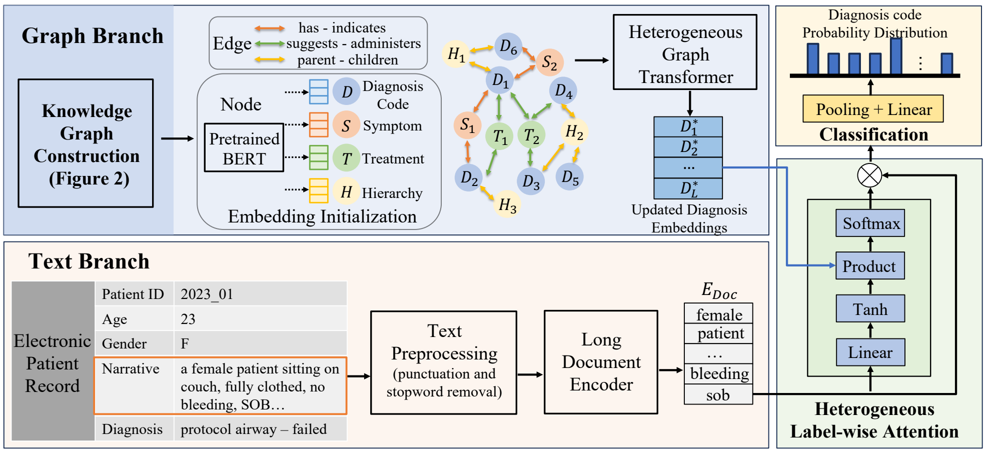


Figure 1: DKEC Pipeline includes three main modules: a text branch to derive text embeddings, a graph branch to derive updated diagnosis embeddings, and a HLA module to derive label-attentive document embeddings.

MIMIC-III dataset and showed low performance for diagnosis prediction (Mullenbach et al., 2018).

Biomedical Knowledge Graph Construction Knowledge graphs provide an efficient way to organize and access the expanding biomedical knowledge. Most existing works (Harnoune et al., 2021; Xu et al., 2020) utilize BERT models to construct biomedical knowledge graphs through named entity recognition and relation extraction. However, BERT is limited by its capacity to process only a fixed number of tokens and is trained for predetermined named-entity classes, making it unsuitable for long biomedical literature with an open set of named-entities. Recent research (Agrawal et al., 2022; Arsenyan et al., 2023; Goel et al., 2023; Hu et al., 2024) shows that LLMs possess excellent zero-shot information extraction capabilities, thus can be suitable for constructing knowledge graphs.

3 Heterogeneous Knowledge Graph Construction

Our goal is to construct a heterogeneous knowledge graph G, which for every disease diagnosis code D_k in a set of *Diagnosis Codes* D: $\{D_k\}_{k=1}^{L}$ (L is the total number of diseases), represents the corresponding sets of medical concepts such as *Signs and Symptoms* S: $\{S_k\}_{k=1}^{|S|}$, *Treatments* T: $\{T_k\}_{k=1}^{|T|}$ and *Hierarchy* : $\{H_k\}_{k=1}^{|H|}$ that have semantic relations with D_k. As shown in Figure 1, the heterogeneous graph of medical concepts is constructed as $G = (N, E)$, with N as the set of nodes and E as the set of edges. There are four different types of nodes in the graph, diagnosis codes D, signs and symptoms S, treatment T, and hierarchy H, and three types of bidirectional edges, has/indicates $\overleftrightarrow{E_{DS}}$ between D and S, suggests/administers $\overleftrightarrow{E_{DT}}$ between D and T, and children/parent $\overleftrightarrow{E_{DH}}$ between D and H. For example, the "Injury - Crush Syndrome" diagnosis code D_i is connected to the signs and symptom "muscle mass" S_j using an edge of type "has/indicates" $\overleftrightarrow{E_{DS}}$. "Injury – Head" ($D_j$) and D_i are the children of ($\overleftrightarrow{E_{DH}}$) node "Adult Trauma Emergencies".

Next, we present a systematic method for automated extraction of disease-relevant medical entities from external knowledge sources and mapping them into normalized concepts for unique representation in the graph. For every diagnosis code in a training dataset, we extract the relevant symptoms and treatments for the disease textual descriptions in online sources and generate the triplets $< D_k, relation, T_k/S_k >$. The hierarchy information is given by the label coding in each dataset.

3.1 Medical Entity Extraction

Online knowledge bases (KBs) are heterogeneous and contain different information on diseases, so we utilize multiple KBs for medical entity extraction (see Figure 2). For a disease diagnosis code D_k, we use the textual description of the disease as the search term to query the KBs. This is done using API calls for KBs with readily available API endpoints and a WebDriver for those without APIs. We then extract the content from the first page of the search results and identify the symptoms and treatments from the text. We evaluated different medical entity extraction methods and used prompting *gpt-4-1106-preview* (Achiam et al., 2023) via one-shot chain-of-thought (one-shot CoT) (Wei et al., 2022) as it showed the best performance (see

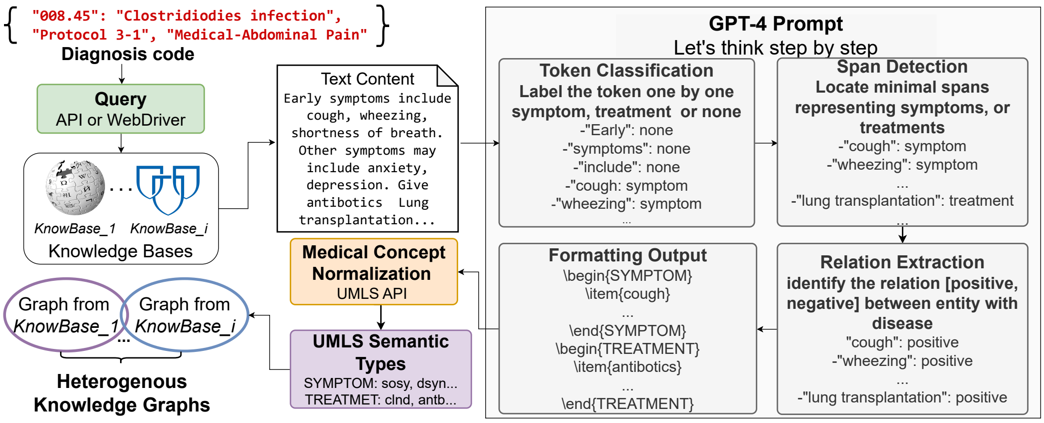


Figure 2: Knowledge Graph Construction

Sections 5.2 and 6.1). As shown in Figure 2, one-shot CoT prompts decompose the task into three sub-tasks and ask GPT-4 to think step by step:
Token Classification asks GPT-4 to label each token in the text by symptom, treatment or none.
Span Detection is the task to locate the minimal span representation of a medical phrase, which is necessary for symptoms like "shortness of breath." This can also help to refine the extracted medical entities better by removing irrelevant modifiers. These two steps are important to prevent GPT-4 from rephrasing medical entities or hallucinations.
Relation Extraction asks GPT-4 to check if the extracted medical phrases from the text are related to the disease to avoid scenarios like negation.

3.2 Medical Entity Representation

For the unique representation of nodes in the graph, we first do *medical concept normalization*. We use the Unified Medical Language System (UMLS) REST API (Bodenreider, 2004) to map the extracted entities presented in different semantic variations to normalized medical concepts. For instance, the entities "fever", "high temperature", and "burning up" can all be mapped to the same UMLS concept "Fever" with Concept Unique Identifier (CUI) C0015967. Specifically, we define semantic type sets for symptoms and treatments (Agrawal et al., 2022; Parwez et al., 2018) and for each entity take the first returned normalized concept with the right semantic type from the API (Appendix A.1).

Then, we generate an *initial node embedding* for each node in the graph by applying a pre-trained BERT model to the node's textual description (see Figure 1) and summing up hidden states in the last four layers for the trade-off between memory and performance (Kenton and Toutanova, 2019).

4 Knowledge-Enhanced Classification

Figure 1 shows the overall DKEC pipeline, consisting of (i) a text branch that extracts features from the input EHR notes using a long document encoder (ii) a graph branch that takes the heterogeneous knowledge graph G as input and incorporates it for label co-occurrence extraction and multi-label classification via a Heterogeneous Label-wise Attention (HLA) mechanism.

4.1 Long Document Encoder

Given that our input is the long text Doc in EHR notes, we need models that can handle temporal sequences and medical terminologies for feature extraction. Prior work (Ji et al., 2022) shows CNNs have superior performance on clinical document classification and pre-trained transformers are limited to encoding a maximum sequence length of 512. To make a fair comparison, we apply DKEC to different state-of-the-art encoders, including multi-filter CNNs and pre-trained transformers.

For multi-filter CNN, similar to work (Zhou et al., 2021), we first map the words in the input text into the low-dimensional word embedding space, then concatenate the convolutional representation from kernel set with different sizes to generate document features $\mathbf{E}_{Doc}$. For BERT models, similar to work (Ji et al., 2021), we chunk the long document into shorter texts and concatenate all the chunked text features from the hidden states in the last layer of the BERT to generate document features $\mathbf{E}_{Doc}$.

4.2 Heterogeneous Label-wise Attention

We use the heterogeneous graph transformer (HGT) (Hu et al., 2020) as the graph model and add another linear layer on the top of HGT's output to derive final embeddings for all the labels (diagnosis nodes). The input of HGT is the initial node embeddings and the medical concept relations, and the output is the updated node embeddings, from which we only use the updated diagnosis embeddings for HLA construction. In the feed-forward phase of the HGT model, a diagnosis node D_k aggregates information from neighboring medical concept nodes S_k, T_k by giving different weights to update itself as $\mathbf{D}_k^\star$. We denote the set of updated diagnosis embeddings from HGT as $\mathbf{D}^\star$: $\{\mathbf{D}_k^\star\}_{k=1}^{L}$,

$$\mathbf{D}^\star = \text{Linear}(\text{HGT}(G)) \tag{1}$$

where $\mathbf{D}^\star \in \mathbb{R}^{L\times\delta}$ is the label representation which incorporates knowledge from diverse medical entities and captures co-occurrence relations in diagnosis codes and δ is a hyper-parameter indicating the dimension of hidden states. We then design an HLA to combine knowledge from each label representation $\mathbf{D}_k^\star \in \mathbf{D}^\star$ with text representation $\mathbf{E}_t$, by having the labels assign different weights for each token in the document representation. The label-wise attention vector is constructed as:

$$\mathbf{a}_{Doc,k} = \text{softmax}(\tanh(\mathbf{W}_0\mathbf{E}_{Doc} + \mathbf{b}_0)\mathbf{D}_k^\star) \tag{2}$$

$$\mathbf{A}_{Doc} = [\mathbf{a}_{Doc,1} \cdots \mathbf{a}_{Doc,k} \cdots \mathbf{a}_{Doc,L}]^T \tag{3}$$

where $\mathbf{W}_0$ and $\mathbf{b}_0$ are respectively the weight and bias of a linear layer to match the size of hidden dimensions in the document representation with the size of label representation and $\mathbf{a}_{Doc,k} \in \mathbb{R}^{\text{Seq}\times n}$ measures how much weight the kth label assigns to each token in document Doc. Finally, we combine all attention vectors $\mathbf{a}_{Doc,k}$ of a document Doc for all L labels to have $\mathbf{A}_{Doc} \in \mathbb{R}^{L\times(\text{Seq}\times n)}$, then the label-wise text representation $\mathbf{E}_{Doc}^{attn} \in \mathbb{R}^{L\times\delta}$ is generated as follows,

$$\mathbf{E}_{Doc}^{attn} = \mathbf{A}_{Doc}\mathbf{E}_{Doc} \tag{4}$$

which measures how informative medical text t is for different labels.

4.3 Classification

The classification layer aims to find the most relevant label $\hat{y}_t$ to the input document Doc. We add another pooling layer for the features obtained from HLA ($\mathbf{E}_{Doc}^{attn}$) before the linear layer to reduce memory usage. The final prediction based on probabilities for each class $\hat{y}_{Doc} \in \mathbb{R}^L$ is achieved after the linear layer:

$$\hat{y}_{Doc} = \text{Linear}(\text{Pooling}(\mathbf{E}_{Doc}^{attn})) \tag{5}$$

The binary cross-entropy loss is applied to measure the distance between each predicted label $\hat{y}_{Doc}$ and ground-truth y_{Doc}.

$$\begin{aligned}\mathcal{L} = -\sum_{Doc=1}^{|Doc|}\sum_{l=1}^{L}&(y_{Doc,l}\log(\hat{y}_{Doc,l}) + \\ &(1-y_{Doc,l})\log(1-\hat{y}_{Doc,l}))\end{aligned} \tag{6}$$

5 Experiments

We conduct extensive experiments to evaluate DKEC by applying it to different baseline language models and comparing its performance to state-of-the-art (SOTA) diagnosis prediction methods. In our experiments, we aim to answer three research questions:
RQ1: Can DKEC improve MLTC performance for class-imbalanced datasets?
RQ2: How does DKEC perform when applied to language models with varying sizes?
RQ3: How does DKEC perform with scaling label sizes?

5.1 Datasets

We used two datasets: a real-world EMS dataset, which is a collection of 4,417 pre-hospital electronic Patient Care Reports (ePCR) annotated with EMS protocol labels, and the benchmark EHR dataset, MIMIC-III (Johnson et al., 2016), which is annotated with ICD-9 diagnosis codes. Both datasets contain textual descriptions of diagnoses, treatment protocols, interventions, and patients' medical histories. Following the pre-processing steps in (Kim et al., 2021), we extract the relevant information from 4,417 ePCRs in the EMS dataset. We use scikit-multilearn (Szymański and Kajdanowicz, 2017) to create 70:30 train/test splits for the EMS dataset and use 10% of the train set for validation. Following the method in (Mullenbach et al., 2018), we split the train, validation, and test sets from MIMIC-III, but we only consider a subset of 3,737 (out of 6,668) ICD-9 diagnosis codes as labels, since knowledge was available for only 3,737 of the codes on the Wikipedia and Mayo Clinic websites. We separate the labels into three categories based on their frequencies in the dataset: *head labels (H)* with more than 1,000 samples, *middle labels (M)* with 10 to 100 samples, and *tail labels (T)* with fewer than 10 samples (few-shot cases). Table 3 shows statistics of the datasets.

	Wikipedia (50 ICD-9 codes)		Mayo Clinic (50 ICD-9 codes)		ODEMSA (43 EMS protocols)	
wo/w NORM	Symptom	Treatment	Symptom	Treatment	Symptom	Treatment
MetaMap	47.62 / 51.53	34.66 / 41.95	44.83 / 49.12	41.82 / 46.44	41.34 / 43.61	39.20 / 41.95
cTAKES	48.74 / 52.58	36.01 / 43.35	42.60 / 46.67	39.67 / 45.35	38.02 / 42.47	48.96 / 52.31
ScispaCy	52.79 / 55.57	41.73 / 49.71	46.54 / 50.43	45.94 / 50.89	44.39 / 47.69	35.88 / 38.82
zero-shot GPT-4	51.99 / 58.77	17.93 / 32.13	52.98 / 63.37	26.16 / 36.48	76.07 / 79.72	10.17 / 23.50
one-shot CoT GPT-4	**84.63 / 86.57**	**85.70 / 89.12**	**82.03 / 86.72**	**90.43 / 93.90**	**86.96 / 91.01**	**86.48 / 88.92**

Table 2: Comparison with baselines on three knowledge bases. *wo/w NORM* means the micro F1-scores are measured before/after medical entity normalization. The best results are highlighted in **bold**.

				N_l		
Dataset	N_{train}	N_{val}	N_{test}	H	M	T
EMS	2787	314	1316	10	21	12
MIMIC-III	47413	1627	3363	494	1038	2205

Table 3: Dataset statistics, N_{train}: number of training instances, N_{val}: number of validation instances, N_{test}: number of test instances, N_l: number of labels in total.

5.2 Knowledge Bases

We constructed two separate heterogeneous graphs for capturing the domain knowledge for ICD-9 diagnosis codes in MIMIC III and protocols in the EMS dataset. For ICD-9 diagnosis codes, **Wikipedia** and **Mayo Clinic** web contents are scraped. For EMS protocols, we use symptom and procedure sections from official EMS guidelines, which are available on the Old Dominion EMS Alliance (**ODEMSA**) website [1]. Statistics of two knowledge graphs are in Appendix A.2. To evaluate the accuracy of different methods for constructing knowledge graphs, we evenly sampled 50 codes from head, middle, and tail classes and manually annotated symptoms and treatments from Wikipedia and Mayo Clinic website contents for ICD-9 diagnosis codes. For EMS protocols, we manually annotated all 43 protocols in ODEMSA documents. The extracted web contents, ground truth annotations, and knowledge graphs are here [2].

Both rule-based and ML-based methods are used as the medical entity extraction baselines, including MetaMap (Aronson, 2001), cTAKES (Savova et al., 2010), ScispaCy (Neumann et al., 2019). The prompt templates for zero-shot and one-shot CoT and baseline model configurations are shown in Appendix A.3. micro F1-score for entity extraction is reported. We count an extracted medical entity as correct if it exactly matches the ground truth.

5.3 Metrics and Parameter Settings

We report the micro F1 (miF) and macro F1 (maF) scores with a fixed threshold of 0.5. miF is heavily influenced by frequent diagnosis codes and thus can be used to evaluate the performance of the head/middle classes. maF weighs the F1 achieved on each label equally and is used to evaluate the performance for the tail classes. Ranking-based metrics (Chalkidis et al., 2019, 2020) recall at k ($R@K$) and precision at k ($P@K$), which do not require a specific threshold, are also reported. $P@K$ is important as it measures the proportion of relevant diagnosis codes suggested in top-k recommendations by the model. $R@K$ is important for medical professionals when they consider the most probable diagnoses for treatments. As the average number of labels per instance in MIMIC-III is 8.0 and EMS is 1.2, we set K as 8 and 1, respectively.

To reduce noise, we did a pre-processing step to remove punctuations and stopwords. We trained each model 5 times, each time with a different random initialization seed. We report the mean ± standard deviation of results with the best parameters. The hidden state size and number of attention heads in graph models are set as 256 and 8, respectively. We use Adam optimizer and regularization with a weight decay of 1e-5 and a dropout rate of 0.2. For training the baselines, we use their best parameter settings. We developed all models by PyTorch (Paszke et al., 2019) and Huggingface (Wolf et al., 2019). All experiments were run on NVIDIA GPU A100 (more details are in Appendix A.4).

5.4 Baselines

We evaluate DKEC in comparison to the following SOTA networks for diagnosis prediction:

Pre-trained Transformers: BERT models including **TinyClinicalBERT**(15M), **DistilBioBERT**(66M) (Rohanian et al., 2023), **COReBERT**(110M) and LLMs like **GatorTron**(325M) and **BioMedLM**(2.7B) are pre-trained on external biomedical knowledge for clinical NLP tasks.

Label-wise Attention Networks: The following baselines were selected due to their superior performance and code availability (Ji et al., 2022):

CAML: The convolutional attention network for multi-label classification (Mullenbach et al., 2018)

[1] https://odemsa.net/

[2] https://github.com/UVA-DSA/DKEC

		Head Labels		Middle Labels		Tail Labels		Overall			
		P@1	R@1	P@1	R@1	P@1	R@1	miF	maF	P@1	R@1
EMS	CAML	$78.6_{\pm1.3}$	$77.7_{\pm1.3}$	$33.0_{\pm0.5}$	$32.6_{\pm0.6}$	$22.7_{\pm4.5}$	$22.7_{\pm4.5}$	$63.7_{\pm1.2}$	$22.4_{\pm1.3}$	$65.0_{\pm1.6}$	$63.5_{\pm1.5}$
	ZAGCNN	$83.0_{\pm1.0}$	$82.0_{\pm1.0}$	$47.0_{\pm1.0}$	$46.2_{\pm0.7}$	$37.9_{\pm7.7}$	$37.9_{\pm7.7}$	$64.8_{\pm1.1}$	$28.3_{\pm2.0}$	$69.6_{\pm0.7}$	$68.1_{\pm0.6}$
	MultiResCNN	$84.3_{\pm0.2}$	$83.2_{\pm0.2}$	$35.6_{\pm1.8}$	$35.0_{\pm2.0}$	$25.0_{\pm2.3}$	$25.0_{\pm2.3}$	$65.8_{\pm0.2}$	$26.1_{\pm0.5}$	$67.9_{\pm0.3}$	$66.3_{\pm0.3}$
	ISD	$81.7_{\pm0.9}$	$80.8_{\pm0.9}$	$44.2_{\pm0.4}$	$43.2_{\pm0.5}$	$29.5_{\pm2.3}$	$29.5_{\pm2.3}$	$67.1_{\pm1.2}$	$26.1_{\pm0.1}$	$68.0_{\pm1.3}$	$66.5_{\pm1.2}$
	GatorTron	$\underline{89.4}_{\pm0.5}$	$\underline{88.4}_{\pm0.5}$	$66.0_{\pm0.4}$	$64.7_{\pm0.7}$	$\underline{57.1}_{\pm2.2}$	$\underline{57.1}_{\pm2.2}$	$75.5_{\pm0.6}$	$35.4_{\pm1.9}$	$77.3_{\pm0.6}$	$75.4_{\pm0.6}$
	BioMedLM	$89.3_{\pm0.3}$	$88.2_{\pm0.3}$	$\underline{71.3}_{\pm0.7}$	$\underline{70.1}_{\pm0.6}$	$47.6_{\pm4.3}$	$47.6_{\pm4.3}$	$\underline{76.9}_{\pm0.7}$	$\underline{43.1}_{\pm1.7}$	$\underline{78.4}_{\pm0.6}$	$\underline{76.6}_{\pm0.6}$
	DKEC-M-CNN	$85.2_{\pm0.7}$	$83.0_{\pm0.7}$	$53.2_{\pm1.3}$	$52.7_{\pm1.1}$	$45.1_{\pm2.1}$	$45.1_{\pm2.1}$	$68.6_{\pm0.4}$	$32.4_{\pm0.6}$	$72.4_{\pm0.4}$	$71.7_{\pm0.6}$
	DKEC-GatorTron	$\mathbf{91.8}_{\pm0.1}$	$\mathbf{90.7}_{\pm0.1}$	$\mathbf{72.4}_{\pm0.4}$	$\mathbf{71.3}_{\pm0.4}$	$\mathbf{67.6}_{\pm2.3}$	$\mathbf{67.6}_{\pm2.3}$	$\mathbf{79.5}_{\pm0.5}$	$\mathbf{51.1}_{\pm1.5}$	$\mathbf{82.2}_{\pm0.5}$	$\mathbf{80.3}_{\pm0.6}$
		P@8	R@8	P@8	R@8	P@8	R@8	miF	maF	P@8	R@8
MIMIC-III	CAML	$54.8_{\pm0.5}$	$57.5_{\pm0.6}$	$5.5_{\pm0.4}$	$28.4_{\pm2.3}$	$0.7_{\pm0.1}$	$4.8_{\pm0.5}$	$51.5_{\pm0.7}$	$4.3_{\pm0.5}$	$54.4_{\pm0.5}$	$50.3_{\pm0.5}$
	ZAGCNN	$55.3_{\pm0.2}$	$58.0_{\pm0.2}$	$6.6_{\pm0.1}$	$34.4_{\pm0.7}$	$1.8_{\pm0.1}$	$11.7_{\pm0.8}$	$52.1_{\pm0.4}$	$4.0_{\pm0.3}$	$55.2_{\pm0.2}$	$51.2_{\pm0.3}$
	MultiResCNN	$\underline{56.5}_{\pm0.3}$	$\underline{59.4}_{\pm0.2}$	$\underline{8.2}_{\pm0.5}$	$\underline{42.3}_{\pm2.8}$	$1.2_{\pm0.1}$	$7.5_{\pm0.9}$	$\mathbf{55.6}_{\pm0.3}$	$\mathbf{6.0}_{\pm0.6}$	$\underline{56.6}_{\pm0.2}$	$\underline{52.7}_{\pm0.2}$
	ISD	$51.8_{\pm0.5}$	$53.8_{\pm0.5}$	$6.1_{\pm0.2}$	$31.7_{\pm1.2}$	$1.9_{\pm0.2}$	$12.6_{\pm0.9}$	$46.8_{\pm1.3}$	$2.8_{\pm0.2}$	$51.6_{\pm0.5}$	$47.5_{\pm0.5}$
	GatorTron	$50.4_{\pm0.2}$	$53.4_{\pm0.2}$	$6.5_{\pm0.2}$	$33.8_{\pm1.1}$	$2.0_{\pm0.3}$	$12.7_{\pm1.4}$	$45.4_{\pm0.4}$	$2.7_{\pm0.3}$	$50.3_{\pm0.2}$	$47.1_{\pm0.2}$
	BioMedLM	$50.5_{\pm0.1}$	$53.4_{\pm0.1}$	$6.1_{\pm0.1}$	$31.3_{\pm1.2}$	$\underline{2.0}_{\pm0.1}$	$\underline{13.2}_{\pm1.1}$	$46.6_{\pm0.3}$	$3.7_{\pm0.5}$	$50.2_{\pm0.1}$	$47.2_{\pm0.2}$
	DKEC-M-CNN	$\mathbf{58.6}_{\pm0.2}$	$\mathbf{61.5}_{\pm0.2}$	$\mathbf{9.6}_{\pm0.1}$	$\mathbf{49.2}_{\pm0.8}$	$2.9_{\pm0.1}$	$\mathbf{19.2}_{\pm0.9}$	$\underline{55.0}_{\pm0.3}$	$4.9_{\pm0.2}$	$\mathbf{58.9}_{\pm0.2}$	$\mathbf{54.8}_{\pm0.2}$
	DKEC-GatorTron	$56.8_{\pm0.4}$	$59.8_{\pm0.2}$	$8.5_{\pm0.1}$	$44.7_{\pm0.7}$	$\mathbf{3.1}_{\pm0.2}$	$19.1_{\pm1.1}$	$53.0_{\pm0.4}$	$\underline{5.7}_{\pm0.3}$	$56.9_{\pm0.4}$	$53.2_{\pm0.3}$

Table 4: Comparison with SOTA on EMS and MIMIC-III (**RQ1**). The best and runner-up results are in **bold** and underlined.

learns attention distribution for each label.

ZAGCNN: Zeroshot attentive GCNN (Rios and Kavuluru, 2018) integrates hierarchical structure of ICD codes by graph CNNs to select label-relevant features for ICD classification.

MultiResCNN: Multi-Filter Residual CNN (Li and Yu, 2020) utilizes a multi-filter convolutional layer to capture n-gram patterns and a residual mechanism to enlarge the receptive field.

ISD: Interactive shared representation network with self-distillation (Zhou et al., 2021) models connections among labels and their co-occurrence.

6 Experimental Results

6.1 Knowledge Graph Quality Evaluation

As shown in Table 2, ***one-shot CoT GPT-4 outperforms other baselines in medical entity extraction*** consistently by a considerable margin. Zero-shot GPT-4 has better performance in extracting symptoms than treatments. In our detailed manual evaluations, we observed that zero-shot GPT-4 usually outputs the whole sentence containing a medical entity or rephrases the medical entities during extraction (see the example in Appendix A.3). With token classification and span detection in one-shot CoT GPT-4 we avoid this problem.

6.2 Class Imbalance Analysis

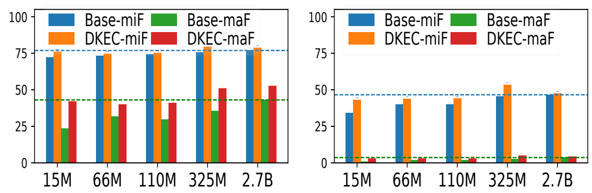
Table 4 shows the performance of DKEC when applied to Multi-filter CNN (DKEC-M-CNN) and GatorTron (DKEC-GatorTron) vs. SOTA on EMS and MIMIC-III datasets. For all the head/middle/tail classes, DKEC outperforms all the baselines. Several observations are highlighted:

DKEC alleviates the class imbalance problem. As shown in Table 4, improvement is most evident on the tail labels. DKEC achieves 10.5% and 6% increase in top-k recall on EMS and MIMIC-III datasets, respectively, compared with runner-up SOTA. On the middle labels, the improvement is still considerable. Compared with runner-up SOTA, DKEC achieves a 6.9% improvement in top-k recall on MIMIC-III. In the head labels where there are sufficient samples, the improvement is relatively small (∼2%). Overall, DKEC maintains a comparable performance to baselines for the head labels while achieving better performance for middle and tail labels, which narrows down the performance gap regardless of data distribution (**RQ1**). We also do an error analysis to understand where and why DKEC underperforms in Appendix 8.

We also observe that transformer models in general achieve a lower performance compared to CNN models on the MIMIC-III dataset, but outperform them on the EMS dataset. This may be due to the different characteristics of the datasets. The EMS dataset contains fewer training samples and labels per sample, while the MIMIC-III notes are longer and each contain a larger number of labels. One hypothesis is that pre-trained transformers perform better on shorter notes and with fewer training samples, which can be further studied in future work.

6.3 Model Size vs. Performance

LLMs have great few-shot abilities but they are costly to train and deploy on resource-constrained devices (Jin et al., 2023; Weerasinghe et al., 2024). So, we apply DKEC to transformers of varying



(a) EMS (b) MIMIC-III

Figure 3: DKEC with different sizes of pre-trained transformers (**RQ2**).

sizes, including LLMs, to answer **RQ2**.

Performance of DKEC-based models increase less as model size grows. Our results show that DKEC is *model-agnostic* and can be applied to different model architectures and sizes from 15M to 2.7B. However, as shown in Figure 3(a) and 3(b), DKEC has more improvement on small language models than LLMs. For example, when applying DEKC, there is a 18.8% improvement in maF over TinyClinicalBERT (15M), while there is only a 9.5% improvement over BioMedLM (2.7B) on the EMS dataset. This might be because LLMs are pre-trained on extensive medical corpora and can handle longer texts, thus show better few-shot abilities. ***DKEC enables smaller language models to achieve comparable performance to LLMs.*** As shown in Figures 3(a) and 3(b), in both datasets, GatorTron (325M) with DKEC outperforms baseline BioMedLM (2.7B) in both miF and maF. This indicates the benefit of DKEC in enabling less costly deployment of small language models in real-world applications.

6.4 Label Size vs. Performance

To understand the effect of label size on the performance of DKEC (**RQ3**), we conduct experiments on subsets of MIMIC-III dataset with varying label sizes, including 1.0k, 3.7k, and 6.7k labels. The knowledge from online sources are fully available for subsets with 1.0k and 3.7k labels, while partially available for the full dataset with 6.7k labels.

As shown in Figure 4, with 6.7k labels, DKEC-M-CNN has similar performance to the best SOTA only with partial knowledge. However, on datasets with 3.7k and 1.0k labels (with full knowledge), DKEC-M-CNN outperforms the best SOTA MultiResCNN by 2∼4% with some memory cost during inference (∼200MB). ***With the increase in the number of labels, the MLTC performance generally drops, but DKEC helps maintain performance, particularly when external knowledge is available for all the labels.*** More results on comparison of DKEC with SOTA with 6.7k and 1.0k labels are available in Appendix A.5.

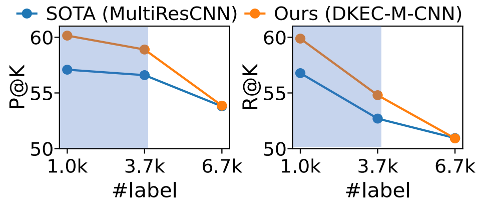


Figure 4: Performance on subsets of MIMIC-III dataset with varying label sizes (**RQ3**). Subsets with 1.0k and 3.7k labels have full knowledge, and 6.7k has partial knowledge.

6.5 Ablation Study

To investigate the effectiveness of DKEC, we conduct an ablation study on the effect of different encoders and label-wise attention mechanisms using MIMIC-III dataset, as shown in Table 5.

Effectiveness of DKEC: For both encoders, incorporating DKEC leads to better performance compared to only using label hierarchy which is the SOTA label-wise attention mechanism.

Effectiveness of External Knowledge: When applying DKEC to M-CNN without hierarchical label relations, performance shows minimal fluctuations. This suggests that incorporating label-specific semantic relations from external knowledge sources is the main driver of performance improvements.

Encoder	Label-wise Attention	miF	maF	P@8	R@8
1-CNN	Label hierarchy*	52.1	4.0	55.2	51.2
1-CNN	DKEC	54.8	4.2	57.5	53.3
GatorTron	Label hierarchy	46.6	3.2	50.7	47.5
GatorTron	DKEC	53.0	**5.7**	56.9	53.2
M-CNN	DKEC	55.0	4.9	**58.9**	**54.8**
M-CNN	DKEC w/o hierarhcy	**55.2**	4.9	58.6	54.5

Table 5: Ablation study using MIMIC-III dataset. "1/M-CNN" are the single/multi-filter CNN. 1-CNN with Label hierarchy* represents the SOTA ZAGCNN.

7 Conclusion

This paper proposes a domain knowledge-enhanced multi-label text classification method for diagnosis prediction. We present an approach for automatic knowledge graph construction from online sources based on medical entity extraction using chain-of-thought prompting with GPT-4. We also introduce a heterogeneous label-wise attention mechanism that incorporates relations among diverse medical entities in the knowledge graph to capture label-related text features for classification. We evaluated our methods on two real-world datasets. Experiments show the accuracy of knowledge graph construction based on three knowledge bases and improved performance over several SOTA methods. We also demonstrated the applicability of our approach to different language models sizes and its scalability to large number of labels.

Limitations

Firstly, we only construct the heterogeneous graphs using two KBs, Wikipedia and Mayo Clinic. This leads to extracting relevant domain knowledge for only a subset of 3,737 diagnosis codes in the MIMIC-III dataset. Larger KBs might be needed to build more complete knowledge graphs. We do not use UMLS as a knowledge base because UMLS interface does not provide direct heterogeneous relations between “disease” and “sign or symptoms” (subset of Finding) and “disease” and “treatments”, which are required for constructing the heterogeneous graphs in this work. Besides, constructing knowledge graphs by UMLS might require extensive manual effort.

Secondly, we manually annotated 50 ICD9 diagnosis codes to illustrate the accuracy of different methods for medical entity extraction. The accuracy of the full knowledge graph would require a considerable amount of human effort.

Lastly, although DKEC performs better than SOTA, more efforts are needed to improve the prediction accuracy of rare diagnosis codes. Given the current SOTA prediction accuracy, particularly for ICD-9 codes in the MIMIC-III dataset, the predictions by DKEC and other baselines studied in this paper should be only used as a reference by healthcare workers and not as a final decision for treatment of the patients in the real world. Further human evaluation of diagnosis prediction models and feedback from medical experts with extensive knowledge of diseases and patient conditions are needed. Large-scale human evaluation and understanding of how the top-k recall and precision results translate to human trust in the system requires more research efforts, which are beyond the scope of this paper.

Acknowledgements

Thanks to the undergraduate researchers, Sion Kim and Saahith Janapati, for their help in preparing the data for this project, and to the anonymous reviewers for their valuable feedback. This work was supported by Award 70NANB21H029 from the U.S. Department of Commerce, National Institute of Standards and Technology (NIST).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Monica Agrawal, Stefan Hegselmann, Hunter Lang, Yoon Kim, and David Sontag. 2022. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alan R Aronson. 2001. Effective mapping of biomedical text to the umls metathesaurus: the metamap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association.

Vahan Arsenyan, Spartak Bughdaryan, Fadi Shaya, Kent Small, and Davit Shahnazaryan. 2023. Large language models for biomedical knowledge graph construction: Information extraction from emr notes. *arXiv preprint arXiv:2301.12473*.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Elliot Bolton, Abhinav Venigalla, Michihiro Yasunaga, David Hall, Betty Xiong, Tony Lee, Roxana Daneshjou, Jonathan Frankle, Percy Liang, Michael Carbin, et al. 2024. Biomedlm: A 2.7 b parameter language model trained on biomedical text. *arXiv preprint arXiv:2403.18421*.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. Hypercore: Hyperbolic and co-graph representation for automatic icd coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3105–3114.

Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2019. Large-scale multi-label text classification on EU legislation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6314–6322, Florence, Italy. Association for Computational Linguistics.

Ilias Chalkidis, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. An empirical study on large-scale multi-label text classification including few and zero-shot labels. *arXiv preprint arXiv:2010.01653*.

Guibin Chen, Deheng Ye, Zhenchang Xing, Jieshan Chen, and Erik Cambria. 2017. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In *2017 International joint conference on neural networks (IJCNN)*, pages 2377–2383. IEEE.

Akshay Goel, Almog Gueta, Omry Gilon, Chang Liu, Sofia Erell, Lan Huong Nguyen, Xiaohong Hao, Bolous

Jaber, Shashir Reddy, Rupesh Kartha, et al. 2023. Llms accelerate annotation for medical information extraction. In *Machine Learning for Health (ML4H)*, pages 82–100. PMLR.

Ayoub Harnoune, Maryem Rhanoui, Mounia Mikram, Siham Yousfi, Zineb Elkaimbillah, and Bouchra El Asri. 2021. Bert based clinical knowledge extraction for biomedical knowledge graph construction and analysis. *Computer Methods and Programs in Biomedicine Update*, 1:100042.

Yan Hu, Qingyu Chen, Jingcheng Du, Xueqing Peng, Vipina Kuttichi Keloth, Xu Zuo, Yujia Zhou, Zehan Li, Xiaoqian Jiang, Zhiyong Lu, et al. 2024. Improving large language models for clinical named entity recognition via prompt engineering. *Journal of the American Medical Informatics Association*, page ocad259.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710.

Shaoxiong Ji, Matti Hölttä, and Pekka Marttinen. 2021. Does the magic of bert apply to medical code assignment? a quantitative study. *Computers in biology and medicine*, 139:104998.

Shaoxiong Ji, Wei Sun, Xiaobo Li, Hang Dong, Ara Taalas, Yijia Zhang, Honghan Wu, Esa Pitkänen, and Pekka Marttinen. 2022. A unified review of deep learning for automated medical coding. *arXiv preprint arXiv:2201.02797*.

Liuyi Jin, Tian Liu, Amran Haroon, Radu Stoleru, Michael Middleton, Ziwei Zhu, and Theodora Chaspari. 2023. Emsassist: An end-to-end mobile voice assistant at the edge for emergency medical services. In *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, pages 275–288.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Sion Kim, Weishi Guo, Ronald Williams, John Stankovic, and Homa Alemzadeh. 2021. Information extraction from patient care reports for intelligent emergency medical services. In *2021 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 58–69. IEEE.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Fei Li and Hong Yu. 2020. Icd coding from clinical text using multi-filter residual convolutional neural network. In *proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8180–8187.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, pages 115–124.

Yang Liu, Hua Cheng, Russell Klopfer, Matthew R Gormley, and Thomas Schaaf. 2021. Effective convolutional attention network for multi-label clinical document classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5941–5953.

Jueqing Lu, Lan Du, Ming Liu, and Joanna Dipnall. 2020. Multi-label few/zero-shot learning with knowledge aggregated from multiple label graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2935–2943, Online. Association for Computational Linguistics.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in Bioinformatics*, 23(6).

Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1903–1911.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1101–1111, New Orleans, Louisiana. Association for Computational Linguistics.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and robust models for biomedical natural language processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Kimberly J O'malley, Karon F Cook, Matt D Price, Kimberly Raiford Wildes, John F Hurdle, and Carol M Ashton. 2005. Measuring diagnoses: Icd code accuracy. *Health services research*, 40(5p2):1620–1639.

Md Aslam Parwez, Muhammad Abulaish, et al. 2018. Biomedical text analytics for characterizing climatesensitive disease. *Procedia computer science*, 132:1002–1011.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: an evaluation of bert and elmo on ten benchmarking datasets. *arXiv preprint arXiv:1906.05474*.

Laila Rasmy, Yang Xiang, Ziqian Xie, Cui Tao, and Degui Zhi. 2021. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. *NPJ digital medicine*, 4(1):86.

Anthony Rios and Ramakanth Kavuluru. 2018. Few-shot and zero-shot multi-label learning for structured label spaces. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2018, page 3132. NIH Public Access.

Omid Rohanian, Mohammadmahdi Nouriborji, Hannah Jauncey, Samaneh Kouchaki, ISARIC Clinical Characterisation Group, Lei Clifton, Laura Merson, and David A Clifton. 2023. Lightweight transformers for clinical natural language processing. *arXiv preprint arXiv:2302.04725*.

Timothy N Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. 2012. Statistical topic models for multi-label document classification. *Machine learning*, 88:157–208.

Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513.

Sile Shu, Sarah Preum, Haydon M Pitchford, Ronald D Williams, John Stankovic, and Homa Alemzadeh. 2019. A behavior tree cognitive assistant system for emergency medical services. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6188–6195. IEEE.

Piotr Szymański and Tomasz Kajdanowicz. 2017. A scikit-based python environment for performing multi-label classification. *arXiv preprint arXiv:1702.01460*.

Betty van Aken, Jens-Michalis Papaioannou, Manuel Mayrdorfer, Klemens Budde, Felix Gers, and Alexander Loeser. 2021a. Clinical outcome prediction from admission notes using self-supervised knowledge integration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 881–893, Online. Association for Computational Linguistics.

Betty van Aken, Jens-Michalis Papaioannou, Manuel Mayrdorfer, Klemens Budde, Felix A Gers, and Alexander Loeser. 2021b. Clinical outcome prediction from admission notes using self-supervised knowledge integration. *arXiv preprint arXiv:2102.04110*.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2020. A label attention model for icd coding from clinical text. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3335–3341. Main track.

Xindi Wang, Robert Mercer, and Frank Rudzicz. 2022. KenMeSH: Knowledge-enhanced end-to-end biomedical text labelling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2941–2951, Dublin, Ireland. Association for Computational Linguistics.

Keshara Weerasinghe, Saahith Janapati, Xueren Ge, Sion Kim, Sneha Iyer, John A Stankovic, and Homa Alemzadeh. 2024. Real-time multimodal cognitive assistant for emergency medical services. *arXiv preprint arXiv:2403.06734*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Jian Xu, Sunkyu Kim, Min Song, Minbyul Jeong, Donghyeon Kim, Jaewoo Kang, Justin F Rousseau, Xin Li, Weijia Xu, Vetle I Torvik, et al. 2020. Building a pubmed knowledge graph. *Scientific data*, 7(1):205.

Xi Yang, Aokun Chen, Nima PourNejatian, Hoo Chang Shin, Kaleb E Smith, Christopher Parisien, Colin Compas, Cheryl Martin, Anthony B Costa, Mona G Flores, Ying Zhang, Tanja Magoc, Christopher A Harle, Gloria Lipori, Duane A Mitchell, William R Hogan, Elizabeth A Shenkman, Jiang Bian, and Yonghui Wu. 2022a. A large language model for electronic health records. *npj Digital Medicine*, 5(1):194.

Xi Yang, Nima PourNejatian, Hoo Chang Shin, Kaleb E Smith, Christopher Parisien, Colin Compas, Cheryl Martin, Mona G Flores, Ying Zhang, Tanja Magoc, et al. 2022b. Gatortron: A large clinical language model to unlock patient information from unstructured electronic health records. *arXiv preprint arXiv:2203.03540*.

Tong Zhou, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Kun Niu, Weifeng Chong, and Shengping Liu. 2021. Automatic icd coding via interactive shared representation networks with self-distillation mechanism.

In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5948–5957.

A Appendix

A.1 UMLS Normalization

We call UMLS API to get top10 ranked medical concepts for a medical entity, and return the first result with a relevant semantic type from our pre-defined set of semantic types for symptoms and treatments. The algorithm is as follows,

Algorithm 1 Medical Entity Normalization

```
Input: Pre-defined Semantic Set $S$, $k$th medical
    entity $e_k$, UMLS API
Output: Normalized Concept $e_k^{norm}$
$r_{k=1}^{10}$ = UMLS($e_k$)  ▷ $r_k$ includes normalized
    concept, CUI, semantic types and etc.
for $k \leftarrow 1$ to 10 do
    if Semantic type of $r_k$ in $S$ then
        $e_k^{norm}$ = normalized concept of $r_k$
        return $e_k^{norm}$
    end if
end for
return null
```

We provide the relevant semantic type set for medical concept normalization.
For *Signs and Symptoms*, the relevant semantic types are "Sign or Symptom (sosy)", "Disease or Syndrome (dsyn)", "Mental or Behavioral Dysfunction (mobd)", "Neoplastic Process (neop)", "Anatomical Abnormality (anab)", "Finding (fndg)", "Pathologic Function (patf)", "Congenital Abnormality (cgab)", and "Injury or Poisoning (inpo)."

For *Treatment*, they are "Therapeutic or Preventive Procedure (topp)", "Antibiotic (antb)", "Clinical Drug (clnd)", "Vitamin (vita)", "Organic Chemical (orch)", and "Amino Acid, Peptide, or Protein (aapp)", "Pharmacologic Substance (phsu)", "Laboratory Procedure (lbpr)", and "Diagnostic Procedure (diap)."

A.2 Knowledge Graph Statistics

The detailed statistics of knowledge graph built for MIMIC-III (3737 labels) and EMS (43 labels) datasets are shown in Table 6. Wikipedia contributes most of unique nodes and edges to the union graph (WoU), and Mayo Clinic website complements some other unique nodes (MoU).

A.3 Medical Entity Extraction Model Configurations

We used the following configurations for MetaMap, cTAKES, ScispaCy and GPT-4.

For MetaMap, we use the Python wrapper pymetamap[3] to perform named entity recognition. The hyperparameters in pymetamap are the semantic types defined in Appendix A.1. For cTAKES, we used the DefaultFastPipeline.piper configuration and then filtered for "textsem: SignSymptomMention". For SciSpacy, we used the Python interface of the model `en_core_sci_scibert`[4] to extract entities and filter by the semantic types defined in Appendix A.1.

For zero-shot GPT-4 and one-shot CoT GPT-4, we provide one example of prompting and its response.
Zero-shot GPT-4: The zero-shot GPT-4 sometimes outputs the whole sentence containing the symptoms or treatments or rephrases them, as shown in the example below.

PROMPT: Given a biomedical text, extract the symptoms and treatments for the disease. The symptom includes signs and symptoms of the disease. The treatment includes medications and procedures used to cure the disease. Remove the dosage of medications. Return your result in latex format as follows,

~~~
```
\begin{SYMPTOM}
\item{Symptom_1}
...
\item{Symptom_n}
\end{SYMPTOM}
\begin{TREATMENT}
\item{Treatment_1}
...
\item{Treatment_n}
\end{TREATMENT}
```
~~~

[...]

RESPONSE:

~~~
```
\begin{SYMPTOM}
~~~

[3] https://github.com/AnthonyMRios/pymetamap

[4] https://allenai.github.io/scispacy/

Dataset	MIMIC-III				EMS
Knowledge Graph	Wikipedia W (WoU)	Mayo Clinic M (MoU)	Combined		ODEMSA
			$W \cup M$	$W \cap M$ (IoU)	
Total Nodes	19275 (0.93)	3672 (0.18)	20835	2112 (0.10)	497
Disease Nodes	3666 (0.98)	320 (0.09)	3737	249 (0.07)	43
Sign and Symptom Nodes	7900 (0.90)	1800 (0.20)	8805	895 (0.10)	310
Treatment Nodes	7709 (0.93)	1552 (0.19)	8293	968 (0.12)	144
Sign and Symptom Edges	37273 (0.92)	3738 (0.09)	40566	445 (0.01)	610
Treatment Edges	42267 (0.94)	3068 (0.07)	44785	550 (0.01)	328

Table 6: Knowledge Graph Statistics. WoU = $\frac{|W|}{|W \cup M|}$; MoU = $\frac{|M|}{|W \cup M|}$; IoU = $\frac{|W \cap M|}{|W \cup M|}$.

~~~
\item{Watery diarrhea three or more
times a day for more than one day}
\item{Mild belly cramping and
tenderness}
[...]
\item{Toxic megacolon}
\item{Sepsis}
\end{SYMPTOM}
\begin{TREATMENT}
\item{Antibiotics}
\item Vancomycin (Firvanq Kit)
\item Fidaxomicin (Dificid)
\item Metronidazole (Flagyl),
used in combination with vancomycin
for severe infection
[...]
\item Probiotics - being researched
for treatment or prevention, not
confirmed for effectiveness outside
of studies
\end{TREATMENT}
```
~~~

One-shot CoT GPT-4

PROMPT: Given a biomedical text, extract the symptoms and treatments for the disease. Follow the four steps in the following example to generate the results. Pay attention to the requirements in every steps. Return your result in latex format.

~~~
```
\begin{SYMPTOM}
\item{Symptom_1}
...
\item{Symptom_n}
\end{SYMPTOM}
\begin{TREATMENT}
\item{Treatment_1}
...
\item{Treatment_n}
\end{TREATMENT}
```
~~~

Here is one example: the disease is Emphysema

TEXT: Early symptoms include cough, wheezing, shortness of breath. Other symptoms may include anxiety, depression. Give antibotics if having bacterial infection. Lung transplantation is an option if having severe lung damage. Do not use antibotic medications like Metronidazole (Flagyl)

Response: Let's think step by step,

Step1: label the tokens one by one "symptom", "treatment" or "none", The "symptom" event type includes signs and symptoms. The "treatment" event type includes medicine and clinical procedures. The "none" event type indicates the token does not belong to "symptom" and "treatment".

```
-Early: none
[...]
-Flagyl: treatment
```

Step2: Refine symptoms and treatments from Step 1 by following criteria,
1.concatenate symptom/treatments spans
2.remove extra irrelevant words in symptoms and treatments

```
-cough: symptom
-wheezing: symptom
-shortness of breath: symptom
-anxiety: symptom
-depression: symptom
-antibotics: treatment
-Lung transplantation: treatment
-Metronidazole: treatment
-Flagyl: treatment
```

Step3: identify the relation [positive, negative] between extracted symptoms/treatments from Step 2 with the disease according to the provided content.

Positive means extracted phrase is the symptom of the disease/treatment improve the disease, negative means extracted phrase is not a symptom of the disease/treatment worsen the disease.
-cough: positive
-wheezing: positive
-shortness of breath: positive
-anxiety: positive
-depression: positive
-antibotics: positive
-Lung transplantation: positive
-Metronidazole: negative
-Flagyl: negative
Step 4: take symptoms/treatments with positive relations with disease from Step 3 and reorganize the result as latex format. Do not include explanation, conditions from Step 3 in your results.

```
\begin{SYMPTOM}
\item{cough}
\item{wheezing}
\item{shortness of breath}
\item{anxiety}
\item{depression}
\end{SYMPTOM}
\begin{TREATMENT}
\item{antibotics}
\item{Lung transplantation}
\end{TREATMENT}
```

Now is the real task: the disease is {title}
TEXT: [...]
Response: Let's think step by step

RESPONSE:

A.4 Training Implementation Details

For all models, we use the same early stopping criteria, namely, if the $R-Precision@K$ does not increase for more than 1e-3 for 10 times, the training is stopped. During the testing, we select the the epoch model that has the best performance. We set the maximum epoch as 200 for CNN-based models. For pre-trained transformers, we set the maximum as 30. This is because pre-trained transformers usually need less epochs to converge while CNN is trained from scratch.

In graph node initialization, we use BlueBERT (Peng et al., 2019) to generate the embedding for DKEC-M-CNN. For pre-trained transformers, we directly use themselves to generate the initial node embeddings. For EMS dataset, we set the HGT layer in graph model as 1 while in MIMIC3 dataset we set the HGT layer as 2. The five random seeds we used are 0, 1, 42, 1234, 3407.

For pre-trained transformers, max-pooling is used before the classification, but for DKEC-M-CNN models, sum-pooling is used instead. Experiments show sum-pooling works best for DKEC-M-CNN, and max-pooling works best for pre-trained transformers. One disadvantage of max-pooling is that it will consume more computation resources than sum-pooling, especially for datasets that have huge amounts of labels. However, there is no clear evidence on which pooling mechanism is optimal. One recommendation is to choose the pooling mechanism based on the data and the need for coding practice. For BioMedLM(2.7B), the FSDP and BFloat16 are applied to speed up training. The last token embedding is used as the document feature for classification.

A.5 Performance on MIMIC-III with different label sizes

As shown in Table 7, we show the performance on MIMIC-III-6668 (all ICD-9 diagnosis codes), and MIMIC-III-1000 (sampled ICD-9 diagnosis codes). Note that For MIMIC-III-6668, only partial labels have domain knowledge while for MIMIC-III-1000, all labels have domain knowledge. And we run all models on the random seed of 3407. All the other settings are the same with as reported in implementation details A.4.

On MIMIC-III-6668 where partial knowledge is available, DKEC-M-CNN has similar overall performance with the best SOTA, with tiny improvement on the tail labels and tiny decrease on the middle labels. On MIMIC-III-1000 where every label has external knowledge (partial knowledge), DKEC-M-CNN outperforms the best SOTA in overall performance and significantly improves the performance in the middle and tail labels.

A.6 Error Analysis

To investigate where DKEC underperforms, we do an error analysis by selecting the best model for each dataset (DKEC-M-CNN for MIMIC-III dataset, and DKEC-GatorTron for EMS dataset) and calculate the FP/FN on head/middle/tail labels.

		Head Labels		Middle Labels		Tail Labels		Overall			
		P@6	R@6	P@6	R@6	P@6	R@6	miF	maF	P@6	R@6
MIMIC-III-1000	CAML	56.1	58.2	6.8	35.8	2.9	17.7	56.0	8.3	55.8	55.1
	ZAGCNN	56.4	58.9	7.4	38.8	2.1	12.8	55.9	9.0	56.2	55.7
	MultiResCNN	57.2	59.9	9.1	47.6	2.6	15.7	59.4	11.5	57.1	56.8
	ISD	54.5	56.2	6.9	36.2	2.0	11.8	54.6	6.9	54.2	53.2
	GatorTron	52.6	54.9	8.4	43.8	3.8	22.6	50.9	6.1	52.4	51.9
	BioMedLM	54.3	57.1	8.9	46.7	5.4	32.4	53.4	7.9	54.3	54.3
	DKEC-M-CNN	**60.2**	**62.9**	11.4	59.2	4.6	27.5	**61.5**	12.2	**60.2**	**59.9**
	DKEC-GatorTron	57.6	60.5	**11.5**	**60.1**	**6.5**	**39.2**	57.7	**13.9**	57.7	57.8
		P@12	R@12	P@12	R@12	P@12	R@12	miF	maF	P@12	R@12
MIMIC-III-6668	CAML	51.3	55.9	4.7	28.1	0.5	4.0	46.4	3.6	51.2	48.2
	ZAGCNN	51.3	55.9	6.1	36.0	1.4	12.1	47.6	3.9	51.5	48.8
	MultiResCNN	53.2	58.1	**7.6**	**44.9**	1.1	9.3	**51.4**	**6.2**	53.8	**51.0**
	ISD	46.3	50.2	5.2	31.0	1.6	13.9	39.0	2.5	46.0	43.5
	GatorTron	43.4	47.9	5.1	30.8	1.2	9.3	37.1	1.8	43.6	41.8
	BioMedLM	44.3	48.9	4.9	29.1	1.2	9.7	39.2	2.4	44.4	42.7
	DKEC-M-CNN	**53.5**	**58.3**	6.9	41.2	1.8	**16.4**	48.7	4.5	**53.9**	50.9
	DKEC-GatorTron	51.7	56.9	6.8	41.0	**1.9**	15.3	46.4	4.3	52.1	49.8

Table 7: Comparison with SOTA on MIMIC-III-1000 and MIMIC-III-6668 datasets. The best result is highlighted in **bold**, and the runner-up is underlined.

Datasets	Head		Mid		Tail	
	FP	FN	FP	FN	FP	FN
EMS	199	128	107	124	4	18
MIMIC-III	4167	15110	131	2796	1	608

Table 8: Numbers of FPs, FNs in head/middle/tail labels

As shown in 8, the number of FNs is much larger than FPs (FN > FP). This indicates that our model is very conservative in decision-making when there are fewer training samples. Further research is needed to alleviate the data imbalance problem.

By manually checking some examples (50 / 253) of model mis-predictions for the EMS dataset, we find two main reasons for the errors. First, as shown in Figure 5, DKEC model is vulnerable to **spurious relations and have no causal reasoning ability** to differentiate the main sign/symptoms for a disease from the secondary ones. It shows the model's inability to analyze the causation between signs/symptoms and diagnosis and just makes decision based on non-relevant words. Future work can focus on debiasing and causal reasoning for improving diagnosis prediction.

Secondly, DKEC is **not effective in distinguishing similar labels**, for example, As shown in Figure 6, in an EHR narrative "difficulty breathing" and "upper respiratory infection" symptoms are mentioned and the ground truth is "medical - respiratory distress/asthma/copd/croup/reactive airway" but the model mispredicted it as "airway - failed". Both of the labels are respiratory system-related problems and confuse the model in decision-making. Similar observations can be found in the case of "childbirth" and "pre-term labor".

Example 1:
ePCR: ...started to have abdominal cramping and vomiting. The patient vomited twice... Abdomen) Soft with abdominal pain to the lower quadrants. 10 out 10 cramping pain...
True Label: "medical - abdominal pain"
Prediction: "medical - nausea/vomiting", "medical - abdominal pain"

Example 2:
ePCR: ...The patient appeared to be working hard to breath with fast rate. C: Difficulty Breathing. H: The patient states that he was late for dialysis yesterday and they did not take enough fluid off ...The patient denied any history of CHF, however did say he is diabetic...
True Label: "airway - failed"
Prediction: "airway - failed", "medical - diabetic - hyperglycemia"

Figure 5: Spurious relation examples. Label-related keywords are in red, and spurious words are in blue.

Example 1:
ePCR: ...The patient was in moderate distress with *difficulty breathing*. The patient appeared to be anxious, and was *breathing at fast rate*. C: *shortness of breath...*
True Label: "medical - respiratory distress/asthma/copd/croup/reactive airway"
Prediction: "airway - failed", "medical - respiratory dis-

tress/asthma/copd/croup/reactive airway"

Example 2:
ePCR: ...Pt was told that she was having *miscarriage*. At the time, pt was months *pregnant*. Pt stated that tonight the *bleeding got significantly worse* and that she has been passing *large clots*...
True Label: "ob/gyn - childbirth/labor/delivery"
Prediction: "ob/gyn - pre-term labor"

Figure 6: Similar label examples